Current Titles in the

Learning with LogoWriter Series

On Your Mark, Get Set, Go

Word Adventures

Words Come Alive!

You and Your Friends

News, Views, and Information

Get the Facts!

And more to come...

LCSI

Logo Computer Systems Inc.
555 W. 57th St., Suite 1236
New York, NY 10019
212 **265 5646**

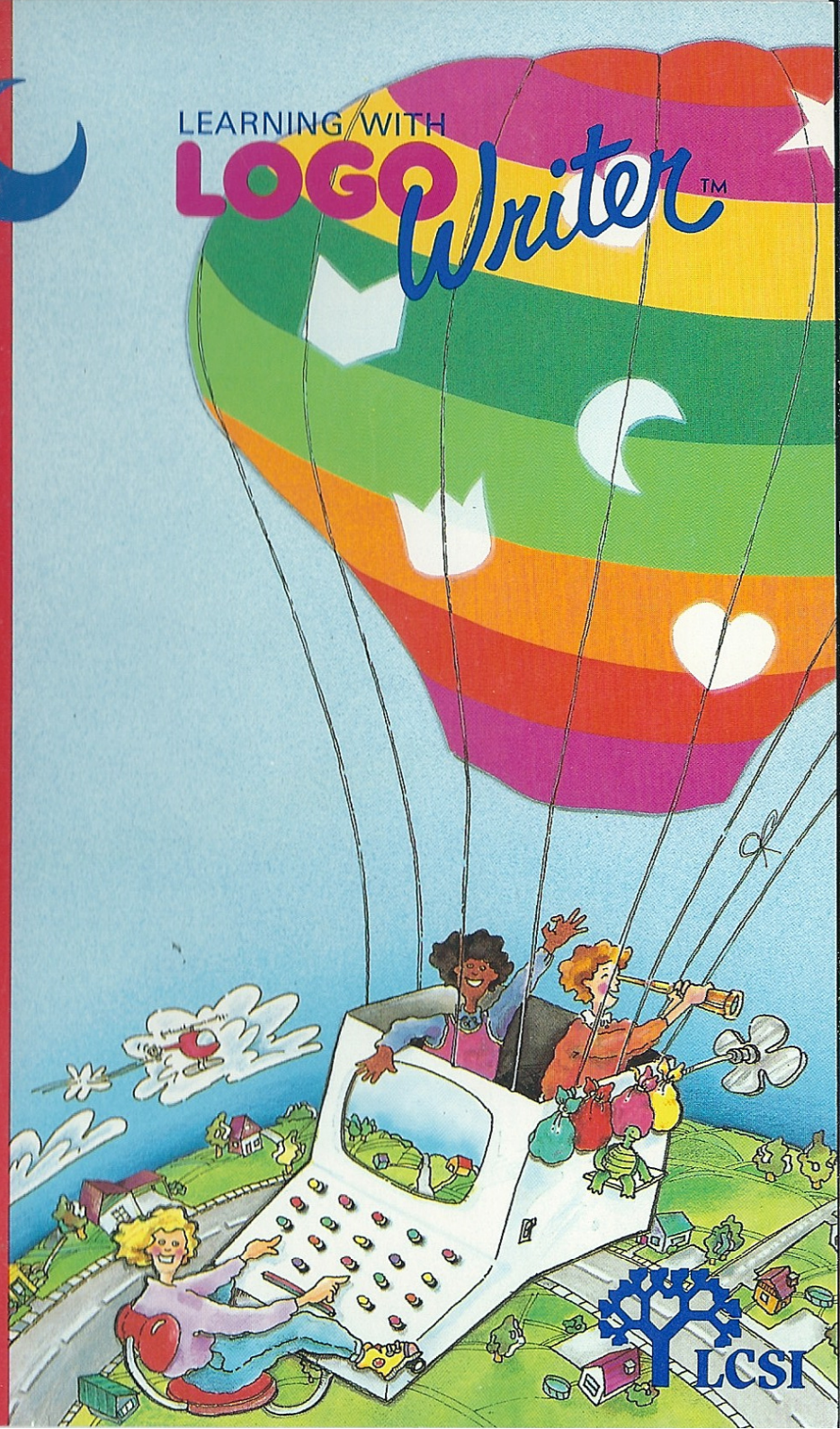
Logo Computer Systems Inc.
9960 Côte de Liesse Road
Lachine, Québec, Canada H8T 1A1
514 **631 7081** – Telex: 05 821624

Printed in Canada

LogoWriter is a Site Licensed Educational Product from Logo Computer Systems Inc.

You and Your Friends

LEARNING WITH LOGOWriter™



LCSI

© Logo Computer Systems Inc. 1986
All rights reserved.

No part of the documents contained herein may be reproduced, stored in retrieval systems or transmitted, in any form or by any means, photocopying, electronic, mechanical, recording or otherwise, without the prior approval in writing from Logo Computer Systems Inc.

Legal deposit, 2nd semester 1986

ISBN 2-89371-004-2
ISBN 2-89371-010-7

LogoWriter is a trademark of Logo Computer Systems Inc.
Apple and the Apple logo are registered trademarks of Apple Computer, Inc.
IBM is a registered trademark and IBM PCjr is a trademark of International Business Machines Corporation.

Learning with LogoWriter

You and Your Friends

by Logo Computer Systems Inc.

ZELLERS ELEMENTARY SCHOOL
STRONGSVILLE PUBLIC SCHOOLS

Montréal New York Tokyo

Table of Contents

Walking Through Your Neighborhood 3
Drawing Your Map 4
Setting Up 6
Walking Along 7
The Tour 8
Try These Ideas 10

More Things to Do 11

How to Copy or Erase Shapes 12

Dice 4 13
Making the Shapes 13
Rolling One Die 14
All Four Dice 15
Rules of the Game 16
Getting Ready to Play 16
Playing the Game 17
Keeping Score 18
Try These Ideas 19

More Things to Do 19

Leaping Turtle 20
Making Your Shapes 20
Drawing the Board 21
Moving 23
Game Rules 24
Checking the Color 25
Playing the Game 27

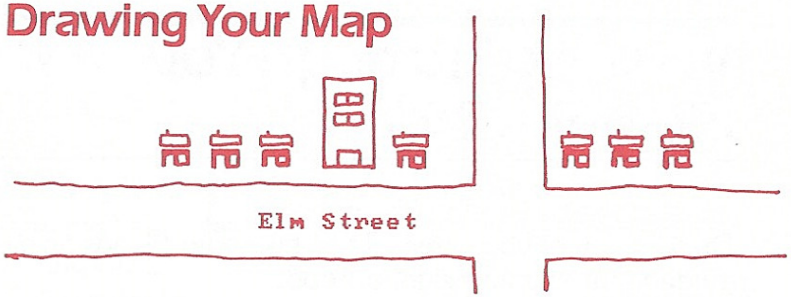
More Things to Do 28

Glossary 28

Walking Through Your Neighborhood

Draw a map of your street. Then take your friends on a guided tour of your neighborhood!

Drawing Your Map



Draw your street. You may want to include the nearest intersection.

Here's a trick.

- Place the turtle on the right side of the page.

```
pu
bk 20
rt 90
fd 100
lt 90
```

- Draw a big rectangle that "wraps" around the page.

```
pd
repeat 2 [fd 160
 rt 90 fd 260 rt 90]
```

To draw the houses on your street you can:

- Define a **house** procedure:

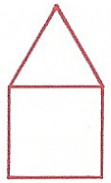
```
to house
repeat 4 [fd 20 rt 90]
fd 20
rt 30
repeat 3 [fd 20 rt 120]
lt 30
bk 20
end
```

This is a good place for a **square** subprocedure.

This is a good place for a **triangle** subprocedure.

Try it:

```
house
```



or

- Set the turtle to the house shape and stamp.

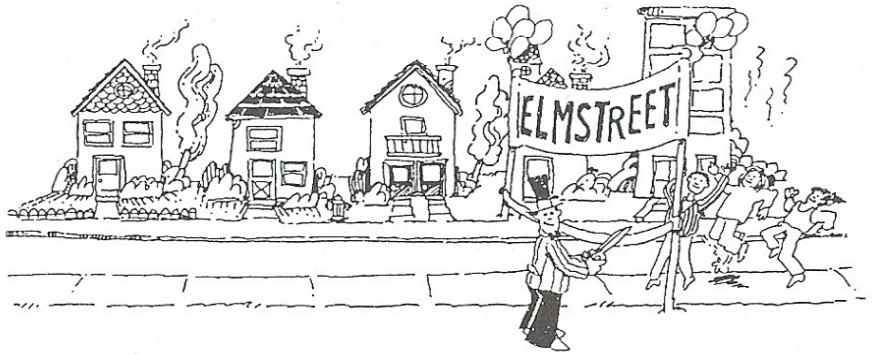
```
setsh 20
pd stamp
```

Draw several houses on your street. Make the houses different shapes, sizes, or colors.

Label your street.

Put your instructions to draw the street into a **street** procedure. Try it:

```
rg
street
```



Setting Up

Give a guided tour of your street.

Make a shape to represent you.

Set the turtle to the shape you created.

Before placing the turtle, send it to its home position:

```
home
```

Now, place the turtle on the first house.

Turn the turtle so it will be heading in the right direction:

```
seth 90
```

Define a **setup** procedure that:

- Sets the turtle's shape.
- Places the turtle at its home position.
- Places the turtle on the first house.
- Sets the turtle's heading.

Try it:

```
rg
street
setup
```


Walking Along

As you give your tour, the person walks from house to house.

Define a **walk** procedure to move the turtle from one house to the next:

```
to walk :distance
repeat :distance [fd 1]
end
```

You can use this procedure to walk different distances. Just change the input to **walk**.

Try it:

```
walk 10
walk 30
```

The Tour

Place the person on the first house again.

Pretend the person is you. To give a guided tour of your street, you would:

- Tell about the first house or the people who live in it.

```
print [This is Sara's house.]
print [She is my friend.]
ct
```

- Walk to the second house.

```
walk ␣
```
Fill in the number.

- Tell about the second house.

```
pr [This is Mario's house.]
pr [He has a big, funny dog.]
ct
```

```
This is Mario's house.
He has a big, funny dog.
```


- Walk to the third house.

```
walk ␣
```
Fill in the number.

- Continue down the street until all the houses have been described.

Define a **tour** procedure. It should:

```
print...   walk...   print...   walk...
```

... and so on.

If the words are cleared too quickly, add a **wait** instruction after each **print** instruction.

Try it out:

```
rg
ct
street
setup
tour
```

Define a **neighborhood** procedure:

```
to neighborhood
rg
ct
street
setup
tour
end
```

Now take your friends for a walk through your neighborhood!

```
neighborhood
```

Add a truck driving down your street.

Tell another turtle to show:

```
tell 1
st
```

Move the turtle onto your street.

Set the turtle to the truck shape and head it in the right direction:

```
setsh 26
seth 90
```

Here is one way to get the truck to move:

```
to move
fd 1
move
end
```

The **move** before **end** tells Logo to do **move** again... *and again... and again... and again...*

Move is a recursive procedure.

To stop the **move** procedure:

Press the **stop** keys ⌘-S | Fn-B | Ctrl-Break

Define a **truck** procedure that:

- Sets turtle 1 to the truck shape.
- Sets turtle 1's heading.
- Places the truck on the street.
- Runs the **move** procedure.

Add **truck** to the end of your **neighborhood** procedure.

```
to neighborhood
rg
ct
street
setup
tour
truck
end
```

More Things to Do

Here are some activity cards for you to do. They all have red tabs.

Climbing a Ladder

A Turtle Eraser

Sound Effects

Table of Contents

Caterpillar

How to Copy or Erase Shapes

To copy a shape:

Use the **next screen** or **previous screen** keys to go to the shape you want to copy.

Press the **copy** keys.

Go to a blank shape.

Press the **paste** keys.

To erase a shape:

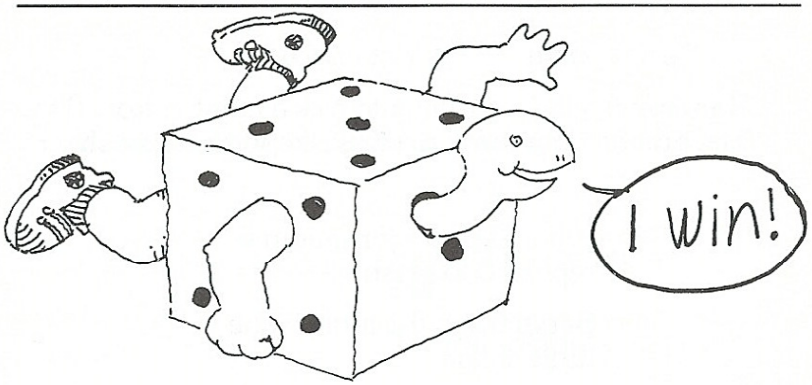
Go to the shape you want to erase.

Press the **cut** keys.

If you change your mind:

Use the **paste** keys to restore the shape.

Dice 4

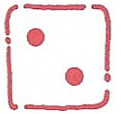


Use turtle dice in a LogoWriter game. Each player rolls the turtle dice and adds up the numbers. The person with the highest score wins.

Making the Shapes

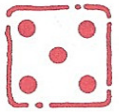
Make six shapes representing the six sides of a die. Use shapes 1 through 6.

shape 1

shape 2 

shape 3

shape 4

shape 5 

shape 6

Rolling One Die

Try this a few times:

```
setsh random 6
```

Random 6 tells LogoWriter to pick a number from 0 to 5. The number that **random** picks is reported to **setsh**.

What happens if **random 6** reports 0 to **setsh**?

Setsh 0 sets the turtle to the turtle shape.

Since you don't want to get 0, add 1 to the number reported by **random 6**.

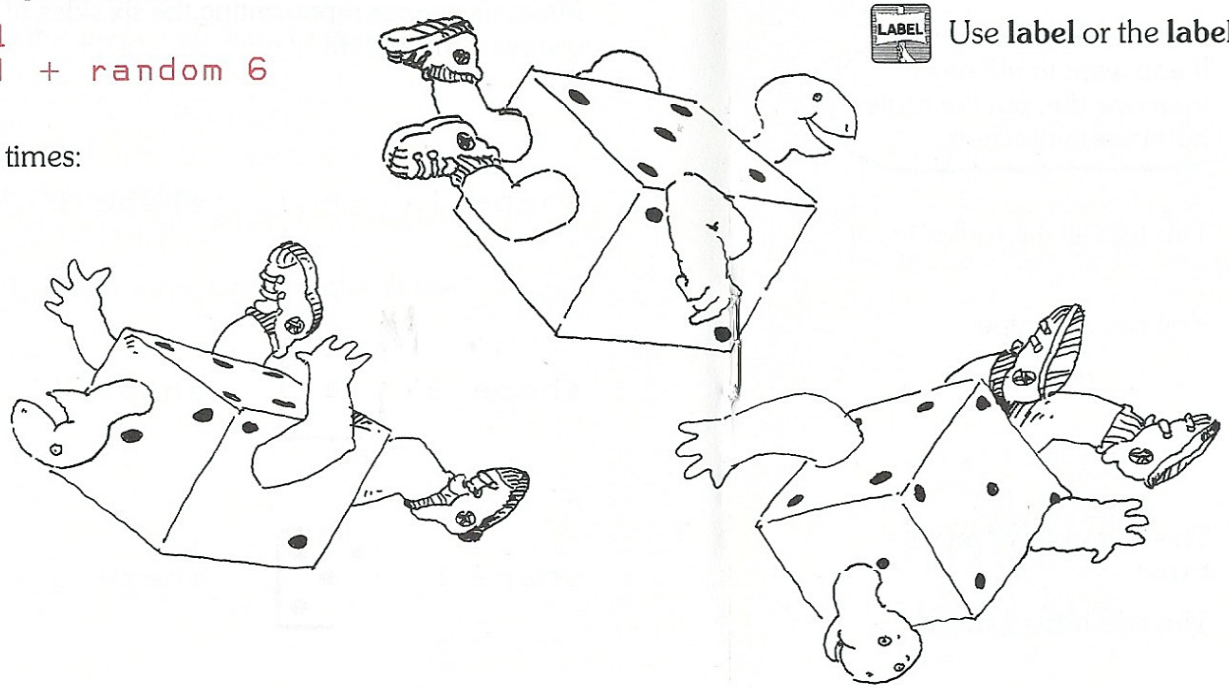
```
setsh 1 + random 6
```

Define a **roll** procedure:

```
to roll
setsh 1 + random 6
end
```

Try it a few times:

```
roll
roll
roll
```



All Four Dice

You will need all 4 turtles for the game.

```
tell all
st
```

All is a reporter. It reports [0 1 2 3] to **tell**.

To set each turtle to a dice shape, tell each of them to **roll**:

```
each [roll]
```

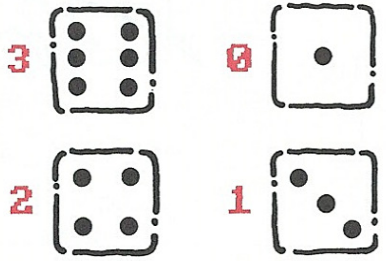
Each gives each turtle a turn to **roll**.

Try **roll** without using **each**:

```
roll
roll
```

If you do not use **each**, all the turtles are set to the same shape.

Use **label** or the **label** keys to type the turtles' numbers.



Rules of the Game

- Each person has 3 turns to roll the turtle dice.
- A player can roll all of the turtle dice or only some of the turtle dice each turn.
- Each player finishes all 3 turns before the next player starts.

Getting Ready to Play

You may want to tell only some of the turtles to roll on your second or third turn.

Define a **play** procedure that talks to only the turtles you want to roll:

```
to play :dice
tell :dice
each [roll]
end
```

:Dice stands for the turtles you want to roll.

If you want to roll more than one die, put the turtle numbers in brackets.

Try it:

```
play [0 1 2 3]
```

This tells all the turtles to roll.

You can also type:

```
play all
```

```
play [2 3]
```

This tells turtles 2 and 3 to roll.

```
play 1
```

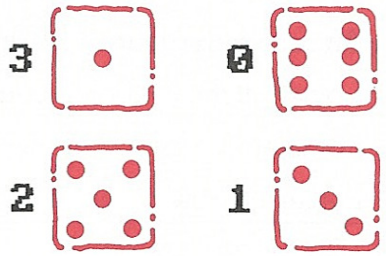
This tells turtle 1 to roll.

Playing the Game

Turn 1

Tell all the turtles to roll.

```
play all
```



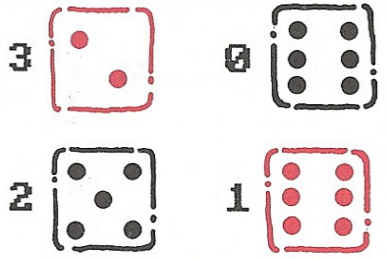
Turn 2

Look at the roll. Keep the turtle dice with the highest numbers.

Roll the other turtles. For example:

```
play [1 3]
```

The player keeps turtles 0 and 2 and tells only turtles 1 and 3 to roll.

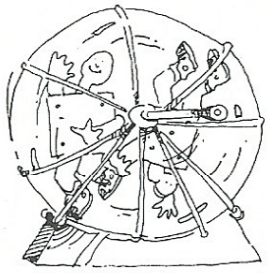


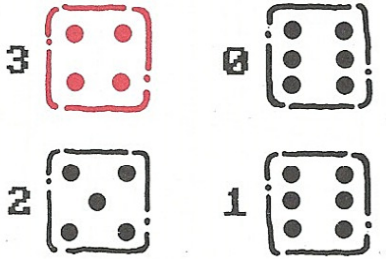
Turn 3

Decide which turtles you want to roll again. For example:

```
play 3
```

The player tells turtle 3 to roll again.





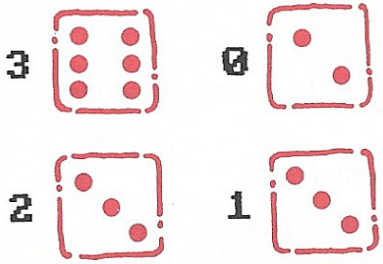
Keeping Score

After your third turn, add up all the numbers on the dice.

LogoWriter can add up your score for you!

For example:

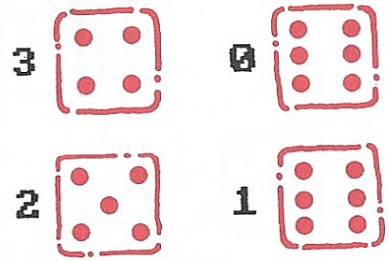
Player 1 throws:



The score is:

```
show 2+3+3+6
14
```

Player 2 throws:




```
show 6+6+5+4
21
```

Player 2 wins!

	0	1	2	3	
Janet	2	3	3	6	14
Mike	6	6	5	4	21

Try these ideas

- Make up different rules for the game:

The player with the lowest score wins.

The player who scores closest to 13 wins.

Play 10 times and add up each person's total scores. The person who has the highest 10 game total wins.

- Make a score sheet with LogoWriter.

More Things to Do

Here are some activity cards for you to do. They all have red tabs.

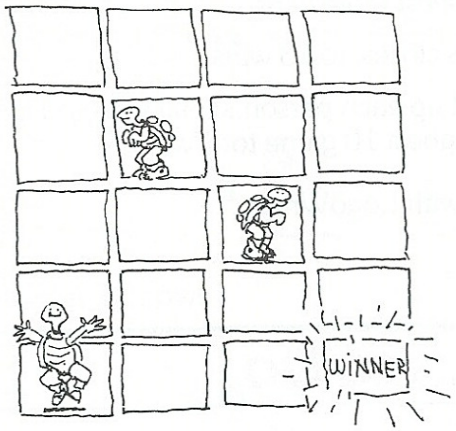
Starry Sky

Quick Change

Big Titles

Leaping Turtle

Create a board game with LogoWriter. If the turtle lands on or passes the winner's square on your turn, you win!!



Making Your Shapes

You need two shapes to set up the game board.

Use the circle shape:

```
shape 12
```

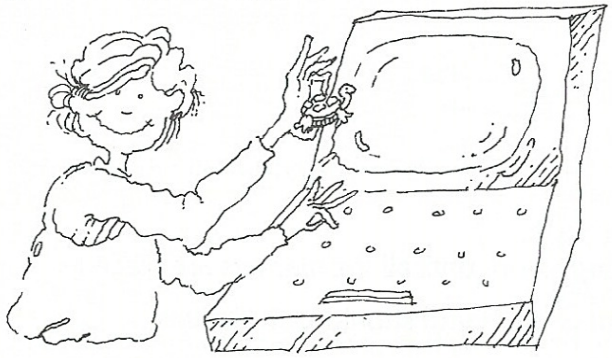
Make another shape that is open in the middle, like a picture frame:

```
shape 7
```

Drawing the Board

Use **setsh**, **setc**, and **stamp** to draw your game board.

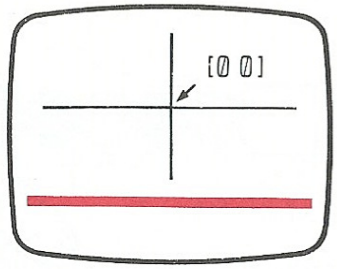
Start by placing the turtle on the left side of the screen, and heading it towards the right:



```
pu
setpos [-130 0]
```

Setpos stands for **set pos**ition. Every position on the screen can be identified by a set of two numbers (coordinates).

The center of the screen is [0 0].




```
seth 90
```

Set the turtle to shape 12. Use this shape for the first 11 markers.

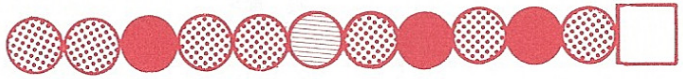
Stamp the open shape at the end of the board.

Each marker should be 20 turtle steps apart.

```
setsh 12
setc 1
repeat 2 [pd stamp pu fd 20]
setc 2
pd
stamp
pu fd 20
setc 1
repeat 2 [pd stamp pu fd 20]
setc 3
pd stamp
```

... and so on, until all the markers are placed.

Your game board should look like this:

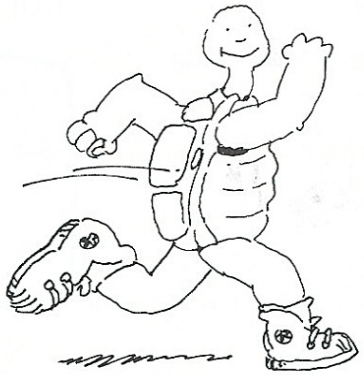


This is the winner's square.

Define a **board** procedure that draws your game board.

Try it:

```
rg
board
```



Moving

Define a **start** procedure that places the turtle at the starting marker:

```
to start
pu
setpos [-130 0]
seth 90
end
```

Set the turtle to the turtle shape and place it on the first marker:

```
setsh 0
start
```

The turtle should move by chance, as if you rolled a die.

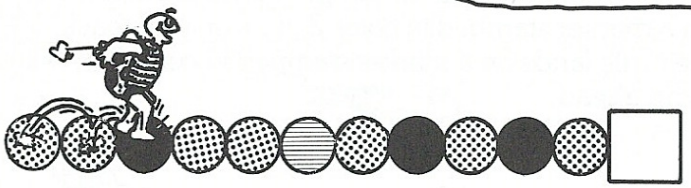
Use **random** to help you.

Try:

```
repeat random 3 [fd 20]
```

Random reports the number it picks to **repeat**.

Each marker is 20 steps apart. **Fd 20** moves the turtle ahead one marker.



If **random 3** picks 2, the turtle jumps ahead 2 markers.

If **random 3** picks 0, the turtle jumps ahead 0 markers.

Define a **move** procedure:

```
to move
repeat random 3 [fd 20]
end
```

Game Rules

Start with these rules:

- If the turtle lands on a marker stamped in color 2, it must jump back two markers.

Define a **jumpback** procedure:

```
to jumpback
repeat 2 [bk 20]
end
```

The **jumpback** procedure runs when the turtle lands on a marker that's stamped in color 2.

- If the turtle lands on a marker stamped in color 3, it will jump ahead three markers.

Define a **jumpahead** procedure:

```
to jumpahead
repeat 3 [fd 20]
end
```

The **jumpahead** procedure runs when the turtle lands on a marker that's stamped in color 3.

- If, when jumping back or jumping ahead, the turtle lands on a marker stamped in color 2, it must jump back. If the turtle lands on a marker stamped in color 3, it will jump ahead.

Checking the Color

How does LogoWriter know when to run each procedure?

Try this:

```
start
if colorunder = 1 [show "Hello]
```

If the color under the turtle is color 1, LogoWriter runs the list of instructions.

Here is a **check** procedure that will check what color is under the turtle:

Stop tells the procedure to stop running.

```
to check
if colorunder = 0
 [pr [You win!!] stop]
if colorunder = 1 [stop]
if colorunder = 2
 [jumpback]
if colorunder = 3
 [jumpahead]
check
end
```

You win! Tells it to stop.

Jumps back.

Jumps ahead. **Check** runs itself again.

Check is a recursive procedure. It contains two stop rules.

```
if colorunder = 0
 [pr [You Win!!]
 stop]
```

This tells **check** to stop running if the color under the turtle is color 0.

```
if colorunder = 1
 [stop]
```

This tells **check** to stop running if the color under the turtle is color 1.

Try it:

```
move
check
```


Playing the Game

Define a **play** procedure that moves the turtle and checks what color marker it landed on:

```
to play
move
check
end
```

Place the turtle on the starting marker:

```
start
```

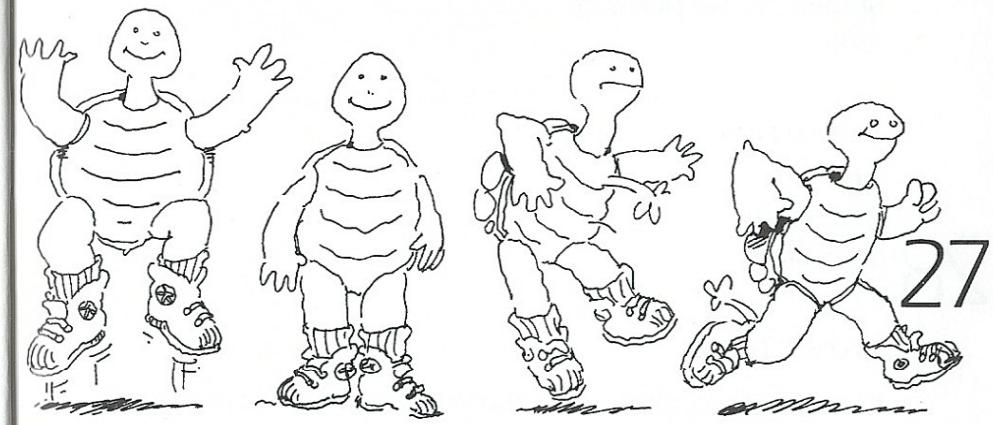
Take turns playing with a friend!

```
play
play
play
play
```

Your turn.
Your friend's turn.
Your turn.
Your friend's turn.

Continue playing.

If the turtle lands on or passes the winner's square on your play, you win!



More Things to Do

Here are some activity cards for you to do. They all have red tabs.

Spirals

Bring Your Page to Life!

A Rainbow

Glossary

Here is a list of the primitives introduced in this booklet:

Commands

each
home
if
setpos (for **set position**)
stop

Reporters

all
colorunder
random

Other Terms

recursive procedure	A procedure that runs itself as a subprocedure.